**Лабораторная работа N6**

Реализуйте следующее Web приложение на фреймворке Flask

1) создайте базу данных на движке SQLite при помощи Python, состоящей из одной таблицы и заполните ее вашими данными (не менее 10 строк),

Проверьте наличие данных в вашей таблице написав SQL запрос через Python SELECT \* FROM ваша\_таблица

2) возьмите файл базы данных созданной в задании 1 и выведите содержимое таблицы в HTML форме, используя Flask фреймворк через браузер

Задание выполнялось на домашнем компьютере с установкой локально Python 3.12

В качестве примера таблицы реализовала список книг, которые я планирую прочесть.

Тест программы по пункту 1 (*создание sql-таблицы и заполнение ее данными*):

import sqlite3

def main():

#соединение с БД

conn= sqlite3.connect('my\_book\_list.db')

#получить курсор

cur= conn.cursor()

#создать таблицу с данными теста, если ее нет

cur.execute('''create table if not exists MyBookList(IdBook integer primary key AUTOINCREMENT,

BookName text,

Author text,

PagesCnt integer,

BookStatus text,

BookBall integer)''')

conn.commit()

#подготовка к вводу вопросов теста

Answer= 'да'

AuthNm= ' '

BookNm= ' '

BookSt= ' '

PagesC= 0

BookBl= 0

#цикл ввода

while Answer == 'да':

BookNm= input('Название книги: ')

AuthNm= input('Автор книги: ')

BookSt= input('Статус (прочитана / в ожидании прочтения): ')

PagesC= int(input('Кол-во страниц: '))

BookBl= int(input('Оценка читателя от 1 до 10 (для неппрочитанных - 0):'))

#сохранить в таблицу

cur.execute('''insert into MyBookList(BookName, Author, PagesCnt, BookStatus, BookBall)

values(?, ?, ?, ?, ?)''',

(BookNm, AuthNm, PagesC, BookSt, BookBl))

conn.commit()

Answer= input('Вы хотите продолжить ввод книг? (да/нет):')

Answer= Answer.lower()

#посмотрим список книг черз Python

cur.execute('select IdBook, BookName, Author, PagesCnt, BookStatus, BookBall from MyBookList order by IdBook')

results= cur.fetchall()

print()

print('Список книг к прочтению:')

print('Номер Название Автор Кол-во стр Статус Оценка')

for Rec in results:

print(f'{Rec[0]:^5} {Rec[1]:35} {Rec[2]:20} {Rec[3]:10} {Rec[4]:15} {Rec[5]}')

#закрыть соединение

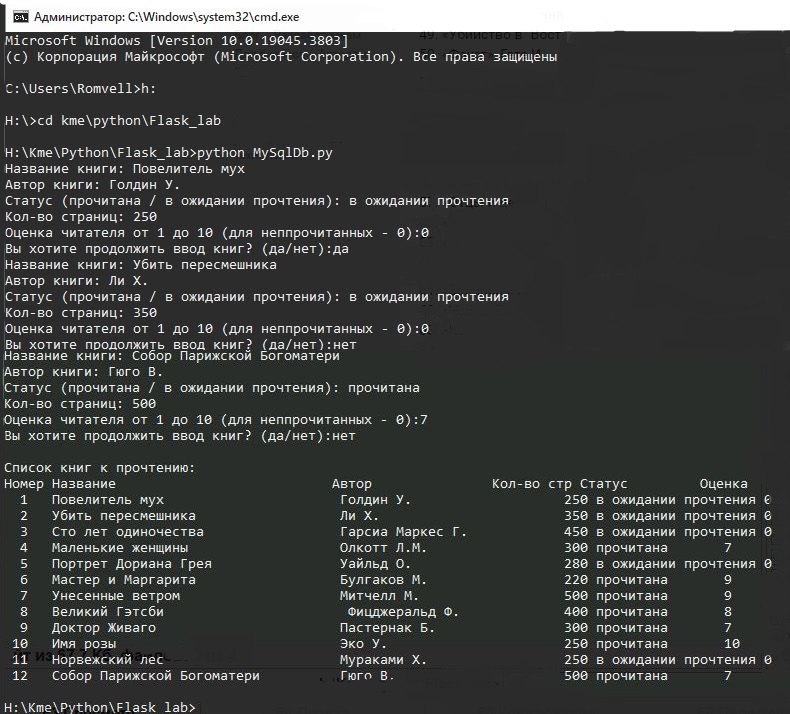
conn.close()

#главная функция

if \_\_name\_\_ == '\_\_main\_\_':

main()

Результат работы программы (склеила начало заполнения таблицы и просмотр результата):



Шаблон html для вывода таблицы (каталог templates):

<!DOCTYPE html>

<html>

<head>

<title>Книги</title>

</head>

<body>

<h1>Лабораторная работа N6</h1>

<table border="1" cellspacing="0" cellpadding="10">

<caption style="font-size: 30px">Список книг, которые я планирую прочесть</caption>

<tr align="center"><th>Название</th><th>Автор</th><th>Кол-во страниц</th><th>Статус</th><th>Оценка</th></tr>

{% for book in books %}

<tr align="left"><th>{{book['BookName']}}</th><th>{{book['Author']}}</th><th>{{book['PagesCnt']}}</th><th>{{book['BookStatus']}}</th><th>{{book['BookBall']}}</th></tr>

{% endfor %}

</table>

</body>

</html>

Тест программы по пункту 2 (вывод данных с использованием Flask):

from flask import Flask, render\_template

import sqlite3

MyApp = Flask(\_\_name\_\_)

@MyApp.route('/')

def index():

conn = sqlite3.connect('my\_book\_list.db')

conn.row\_factory = sqlite3.Row

books = conn.execute('select \* from MyBookList order by BookStatus ').fetchall()

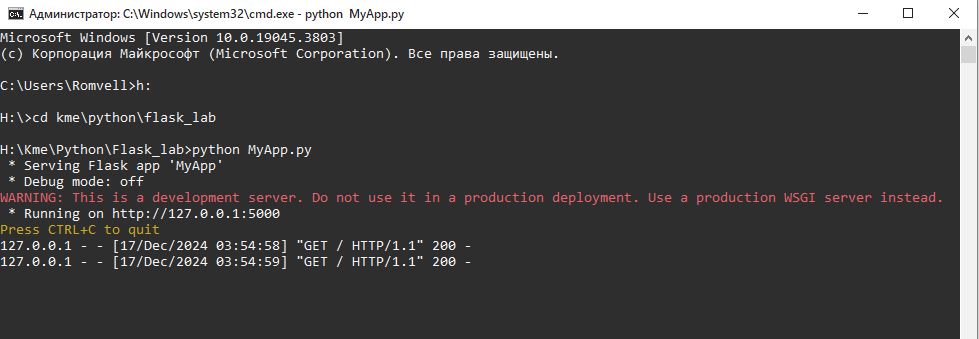
conn.close()

return render\_template('booklst.html', books=books)

if \_\_name\_\_ == '\_\_main\_\_':

MyApp.run()

Результат работы программы (запуск серверной части):



Результат работы программы (страница браузера):

